

APPENDIX

DATA BANK DESCRIPTION OF THE VARIOUS UNITS OF THE BACK END 20

[00148] This appendix provides an exemplary listing of various back-end units.

COMPANY UNIT

[00149]

```

DROP      SEQUENCE      seq__dapt__company;
DROP      TABLE      dapt__company;
CREATE      TABLE      dapt__company

      id              INT4  PRIMARY KEY,
      name            TEXT  NOT NULL,
      street1         TEXT, NOT NULL,
      street2         TEXT,
      city            TEXT  NOT NULL
      state           TEXT,
      zip             INT4  NOT NULL
      country         TEXT  NOT NULL
      phone           TEXT  NOT NULL
      fax             TEXT,
      email           TEXT,
      www             TEXT,
      resources       TEXT,
      contractSequence INT4  DEFAULT(0),

      UNIQUE (name, street1, zip, city)
);
CREATE      SEQUENCE      seq__dapt__company;
```

[00150] Company addresses can be stored in the data bank. These company addresses can only be viewed, modified, compiled or deleted by the ASP administrator.

ACCESS RIGHTS UNIT

[00151]

```

DROP      TABLE dapt__permission;
CREATE      TABLE dapt__permission
(
      id              INT4  PRIMARY KEY,
      remark          TEXT  NOT NULL
);
```

```

INSERT INTO dapt__permission (id,remark) VALUES (0,'ASP Administrator');
INSERT INTO dapt__permission (id,remark) VALUES (1,'Company Administrator');
```

```
INSERT INTO dapt_permission (id,remark) VALUES (2,'Customer Advisor');
INSERT INTO dapt_permission (id,remark) VALUES (3, 'Customer User');
```

EMPLOYEE UNIT

[00152]

```
DROP      SEQUENCE      seq_dapt_employee;
DROP      TABLE      dapt_employee;
CREATE    TABLE dapt_employee
(
    id                                INT4    PRIMARY KEY,
    firstName                        TEXT NOT NULL,
    lastName                        TEXT NOT NULL,
    title                          TEXT ,
    gender_id                       INT4 REFERENCES dapt_gender(ID) NOT NULL,
    dateOfBirth                     DATE NOT NULL,
    company_id                      INT4 REFERENCES dapt_company(id) NOT NULL,
    ident                           TEXT ,
    phone                           TEXT NOT NULL,
    fax                             TEXT ,
    email                           TEXT ,
    www                             TEXT ,
    username                        TEXT NOT NULL,
    password                        TEXT NOT NULL,
    permission_id                   INT4 REFERENCES dapt_permission NOT NULL,
    create                          DATE DEFAULT CURRENT_DATE,
    lastLogin                       INT4 ,
    sessionKey                      TEXT ,
    substitute_id                   INT4 DEFAULT(-1),
    jobTitle                        TEXT,
    UNIQUE( company_id, username)
);
CREATE    SEQUENCE      seq_dapt_employee;
```

[00153] Each employee is allocated to exactly one company in the company unit (*company_id*). What this allocation controls is that only customer datasets that belong to this company are visible. Furthermore, the employee is equipped with access rights (*permission_id*) (see the above description). These access rights control what information the employee is offered by the system. Thus, for example, an account manager (*permission_id*=3) cannot view any statistics above the

company itself (number of contracts, number of employees, contracts per employee, etc.). In order to enable employees to be represented, a reference to another employee is also maintained. As soon as an employee logs off and defines a representative, all information are also made available to the representative.

[00154] For authentication, every employee is assigned a user name as well as a password. Upon logon to the system, the user is requested to enter the company name as well as user name and the password. The server application checks based on these parameters whether there is a user belonging to the indicated company and subsequently verifies the password. When these unambiguous parameters agree, the employee can be identified. A session key is generated in the data bank that unambiguously identifies the logged on user.

BENEFICIARIES

[00155]

```

DROP      TABLE dapt_beneficiary __mode;
CREATE      TABLE dapt __beneficiary __mode
(
    id      INT4 PRIMARY KEY,
    remark  TEXT NOT NULL
);

```

```

INSERT INTO dapt_beneficiary__mode (id,remark) VALUES (0, 'SINGLE' );
INSERT INTO dapt_beneficiary__mode (id,remark) VALUES (1, 'PAIR' );
INSERT INTO dapt_beneficiary__mode (id,remark) VALUES (2, 'COMPANY' );

```

```

DROP      SEQUENCE      seq_dapt_beneficiary;
DROP      TABLE      dapt_beneficiary;
CREATE      TABLE      dapt_beneficiary
(
    id      INT4 PRIMARY KEY,
    mode__id  INT4 REFERENCES dapt_customer_mode(id)DEFAULT(0),
    firstName  TEXT NOT NULL,
    lastName  TEXT NOT NULL,
    title  TEXT,
    gender_id  INT4 REFERENCES dapt_gender(ID) NOT NULL,
    dateOfBirth  DATE NOT NULL,
    firstName2  TEXT ,
    lastName2  TEXT ,
    title2  TEXT ,

```

```

gender_id2    INT4 REFERENCES dapt_gender(ID) DEFAULT(0),
dateOfBirth2  DATE DEFAULT CURRENT_DATE,

accountNumber INT4 NOT NULL,
bankIdentificationCode NOT NULL,
bankDetail    TEXT NOT NULL,
company_id    INT4 REFERENCES dapt_company(id) NOT NULL
);

CREATE          SEQUENCE          seq_dapt_beneficiary;

```

[00156] Payouts of investments can generally be paid out to different accounts (of the beneficiary). It is thus not only a customer but also a beneficiary that is assigned to each contract.

CUSTOMER UNIT

[00157]

```

DROPTABLE dapt_customer_status;
CREATE     TABLE dapt_customer_status
(
    id          INT4 PRIMARY KEY,
    remark      TEXT NOT NULL
);

```

```

INSERT INTO dapt_customer_status (id,remark) VALUES (0, 'REGISTERED'
);
INSERT INTO dapt_customer_status (id,remark) VALUES (1, 'CONFIRMED'
);
INSERT INTO dapt_customer_status (id,remark) VALUES (2, 'CHECKOUT'
);
INSERT INTO dapt_customer_status (id,remark) VALUES (3,
'CHECKOUTCONFIRMED' );
INSERT INTO dapt_customer_status (id,remark) VALUES (4, 'CHANGED');

```

```

DROPTABLE dapt_customer_mode;
CREATE     TABLE dapt_customer_mode
(
    id          INT4 PRIMARY KEY,
    remark      TEXT NOT NULL
);

```

```

INSERT INTO dapt_customer_status (id,remark) VALUES (0, 'SINGLE' );
INSERT INTO dapt_customer_status (id,remark) VALUES (1, 'PAIR' );
INSERT INTO dapt_customer_status (id,remark) VALUES (2, 'COMPANY' );

```

```

DROP          SEQUENCE          seq_dapt_customer;

```

```

DROP      TABLE      dapt_customer;
CREATE    TABLE      dapt_customer;
(
    id          INT4    PRIMARY KEY,
    mode_id     INT4    REFERENCES dapt_customer_mode(id) DEFAULT(0),
    firstName   TEXT    NOT NULL,
    lastName    TEXT    NOT NULL,
    title       TEXT    ,
    gender_id   INT4    REFERENCES dapt_gender(ID) NOT NULL,
    dateOfBirth DATE    NOT NULL,

    firstName2  TEXT    ,
    lastName2   TEXT    ,
    title2      TEXT    ,
    gender_id2  INT4    REFERENCES dapt_gender(ID) DEFAULT(0),
    dateOfBirth2 DATE    DEFAULT CURRENT_DATE,

    company_id  INT4    REFERENCES dapt_company(id) NOT NULL,
    team_id     INT4    REFERENCES dapt_team(id)   NOT NULL,
    status_id   INT4    REFERENCES dapt_customer_status(id) NOT
NULL
    street1     TEXT    NOT NULL,
    street2     TEXT    ,
    city        TEXT    ,
    state       TEXT    ,
    zip         INT4    NOT NULL,
    country     TEXT    NOT NULL,
    phone       TEXT    ,
    fax         TEXT    ,
    email       TEXT    ,
    www         TEXT    ,

    accountNumber INT4  DEFAULT(-1),
    bankIdent     INT4  DEFAULT(-1),
    bankDetail    TEXT  ,
    creator_id    INT4  REFERENCES dapt_employee(id) NOT NULL,
    createDate    DATE  NOT NULL,
    createTime    TIME  NOT NULL,

    modifier_id  INT4  REFERENCES dapt_employee(id) NOT NULL,
    modifiedDate DATE  NOT NULL,
    modifiedTime TIME  NOT NULL
);

```

```

CREATE      SEQUENCE  seq_dapt_customer;

```

UNIT FOR STORING FUND COMPANIES

[00158]

```

DROP      SEQUENCE seq_dapt_company_address;
DROP      TABLE dapt_company_address;
CREATE    TABLE      dapt_company_address

```

```

    id          INT4 PRIMARY KEY,
    company_id  INT4 REFERENCES dapt_company(ID) NOT NULL,
    name        TEXT NOT NULL,
    street1     TEXT NOT NULL,

    street2     TEXT,
    city        TEXT NOT NULL,
    state       TEXT,
    zip         INT4 NOT NULL,
    country     TEXT NOT NULL,
    phone       TEXT NOT NULL,
    fax         TEXT,
    email       TEXT,
    www         TEXT,
    remark      TEXT,
    creator_id  INT4, REFERENCES dapt_employee(id) NOT NULL,
    createDate  DATE NOT NULL,
    createTime  TIME NOT NULL,

    modifier_id INT4 REFERENCES dapt_employee(id) NOT NULL,
    modifiedDate DATE NOT NULL,
    modifiedTime TIME NOT NULL

```

);

```

CREATE      SEQUENCE seq_dapt_company_address;

```

DEPOSIT UNIT

[00159]

```

DROP SEQUENCE seq_dapt_deposit;
DROP TABLE      dapt_deposit;
CREATE TABLE      dapt_deposit

```

```

(
    id          INT4 PRIMARY KEY,
    customer_id INT4 REFERENCES dapt_customer(id) NOT
NULL,
    name        TEXT NOT NULL,
    deposit AccountNumber INT4 NOT NULL,
    bankIdentificationCode INT4 NOT NULL,
    bankDetail   TEXT NOT NULL,
    UNIQUE (customer_id, depositAccountNumber, bankIdentification Code)

```

);

```

CREATE      SEQUENCE seq_dapt_deposit;

```

[00160] Arbitrary deposits can be assigned to each customer. The deposits are usually requested by the customer himself in writing in parallel (apart from the system). These written documents are available to the consultant. Securities are booked in this deposit.

FEE UNIT

[00161]

```
DROP SEQUENCE seq_dapt_fee;
DROP TABLE dapt_fee;
CREATE TABLE dapt_fee
(
    id INT4 PRIMARY KEY
    company_id INT4 REFERENCES dapt_company(id) NOT NULL,
    name TEXT NOT NULL,
    perAnnum INT4 NOT NULL,
    fee FLOAT NOT NULL,
    feeMinimum FLOAT NOT NULL

    creator_id INT4 REFERENCES dapt_employee(id) NOT NULL,
    createDate DATE NOT NULL,
    createTime TIME NOT NULL,

    modifier_id INT4 REFERENCES dapt_employee(id),
    modifiedDate DATE,
    modifiedtime TIME,

    UNIQUE (name, fee, feeMinimum, perAnnum, company_id)
);
CREATE SEQUENCE seq_dapt_fee;
```

[00162] Fee models are allocated to each company. These fee models can contain minimum amounts and can also be marked as fees to be paid annually. Every securing contract is linked to a fee model.

CONTRACT UNIT

[00163]

```
DROPTABLE dapt_return_mode;
CREATE TABLE dapt_return_mode
(
    id INT4 PRIMARY KEY,
```

```

        remark      TEXT NOT NULL
);

```

```

INSERT INTO dapt_return_mode(id,remark) VALUES (0,'ABSOLUTE');
INSERT INTO dapt_return_mode(id,remark) VALUES (1,'RELATIVE');

```

```

DROP TABLE dapt_contract_status;
CREATE TABLE dapt_contract_status
(

```

```

        id          INT4 PRIMARY KEY,
        remark      TEXT NOT NULL
);

```

```

INSERT INTO dapt_contract_status (id,remark) VALUES (0,'REGISTERED' );
INSERT INTO dapt_contract_status (id,remark) VALUES (1,'CONFIRMED' );
INSERT INTO dapt_contract_status (id,remark) VALUES (2,'ACTIVATED' );
INSERT INTO dapt_contract_status (id,remark) VALUES (3,'MONITORED' );
INSERT INTO dapt_contract_status (id,remark) VALUES (4,'TERMINATED' );
INSERT INTO dapt_contract_status (id,remark) VALUES (5,'CHECKOUT' );
INSERT INTO dapt_contract_status (id,remark) VALUES (6,'CHECKOUTCONFIRMED');

```

```

DROP SEQUENCE seq_dapt_team;
DROP TABLE dapt_team;
CREATE TABLE dapt_team
(

```

```

        id          INT4 PRIMARY KEY,
        name        TEXT      DEFAULT('--'),
        advisor_id  INT4      REFERENCES dapt_employed(id) NOT NULL
        user_id     INT4      REFERENCES dapt_employed(id) NOT NULL
        UNIQUE (advisor __id, user __id)
);

```

```

CREATE SEQUENCE seq_dapt_team;

```

```

DROP SEQUENCE seq_dapt_contract;
DROPTABLE dapt_contract;
CREATE TABLE dapt_contract
(

```

```

        id          INT4 PRIMARY KEY,
        deposit_id  INT4 REFERENCES dapt_deposit(id) NOT NULL,
        nsin        INT4 NOT NULL
        reference    INT4 NOT NULL,

        beneficiary_id INT4 REFERENCES dapt_beneficiary(id) NOT NULL,
        broker_id     INT4 REFERENCES dapt_team(id) NOT NULL,
        company_id    INT4 REFERENCES dapt_company(id) NOT NULL,
        speculativePeriod INT4

```

```

        numberAssets FLOAT CHECK (numberAssets>=0),

```


purchaseDate	DATE NOT NULL,
purchaseTime	TIME NOT NULL,
purchasePrice	FLOAT NOT NULL CHECK (price>=0),
activation	FLOAT,
activationMode__id	INT4 REFERENCES dapt__return__mode,
activated	INT4
activationDate	DATE NOT NULL,
activationTime	TIME NOT NULL,
safeGuard	FLOAT NOT NULL,
safeGuardMode__id	INT4 REFERENCES dapt__return__mode(id),
safeGuardFee__id	INT4 REFERENCES dapt__fee(id) NOT NULL,
StoppLoss	FLOAT NOT NULL,
StoppLossMode__id	INT4 REFERENCES dapt__return__mode(id),
StoppLossFee__id	REFERENCES dapt__fee(id),
price	FLOAT NOT NULL,
priceDate	DATE NOT NULL,
priceTime	TIME NOT NULL,
threshold	FLOAT NOT NULL,
thresholdDate	DATE NOT NULL,
thresholdTime	TIME NOT NULL,
status__id	INT4 REFERENCES dapt__contract__status
DEFAULT(0),	
statusDate	DATE NOT NULL,
statusTime	TIME NOT NULL,
statusEmployee__id	INT4 REFERENCES dapt__employee(id) NOT NULL,
systemReference	INT4 NOT NULL,
systemDate	DATE NOT NULL,
systemTime	TIME NOT NULL,
volume	FLOAT NOT NULL,
salePrice	FLOAT,
saleDate	DATE,
saleTime	TIME,
contractNumber	INT4 DEFAULT (-1)

);

CREATE SEQUENCE seq__dapt__contract;

RATE DATA UNIT

[00164]

```
DROP SEQUENCE seq_dapt_quote;  
DROP TABLE dapt_quote;
```

```
CREATE TABLE dapt_quote  
(  
    id          INT4 NOT NULL,  
    nsin        INT4 PRIMARY KEY,  
    reference   INT4 NOT NULL,  
    name        TEXT NOT NULL,  
  
    price       FLOAT NOT NULL CHECK (price>=0),  
    date        DATE NOT NULL,  
    time        TIME NOT NULL,  
  
    UNIQUE (id, nsin, reference)  
);
```

```
CREATE SEQUENCE seq_dapt_quote;
```

NOTIFICATION UNIT

[00165]

```
DROP          SEQUENCE seq_dapt_notification;  
DROP          TABLE dapt_notification;
```

```
CREATE TABLE dapt_notification  
(  
    id          INT4 PRIMARY KEY,  
    target_id   INT4 REFERENCES dapt_permission(id) NOT NULL,  
    company_id  INT4 REFERENCES dapt_company(id) NOT NULL,  
    message     TEXT NOT NULL,  
    link        TEXT NOT NULL,  
    date        DATE NOT NULL,  
    time        TIME NOT NULL  
);
```

```
CREATE SEQUENCE seq_dapt_notification;
```

[00166] Account managers and consultants can themselves activate contracts or customer applications without having to wait for a confirmation of the company

administrator. The company administrator is notified of this bypassing of the actual work sequence with a message.